Received: March 17, 2017

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Double QoS Implementation in the Network Bandwidth Adjustment Task

Guchenko Mykola[1] Sokhin Natalia[1] Lugovyj Oleksandr[1]
Dobiáš Václav[2*] Čejka Jiří[2] Ludvík Miroslav[2]

[1]*Kremenchuk Mykhailo Ostrogradskyi National University, Ukraine*
[2]*The Institute of Technology and Business in České Budějovice, Czech Republic*
* Corresponding author's Email: dobias@mail.vstecb.cz

Abstract: The actual task is to research QoS (Quality of Service) instruments in order to improve existing and develop new methods. This paper presents the modified method of double QoS, which differs from standard one by application of multi-level traffic analysis by user groups, types of traffic and priorities. The efficiency of the proposed method is proved experimentally on routers with the Mikrotik operating system by means of local computer network simulation, based on virtual machines. The experiments were conducted with generated multi-traffic with a uniform distribution for Packet Size and Internet Departure Time parameters. Compared results of router work with standard and modified methods. Using the proposed method allows reducing the number of dropped packets by simply configuring the router system tools and thus improving the quality of user service.

Keywords: Double QoS, Virtual machine, Simulation model, Packets loss.

1. Introduction

In modern telecommunication networks the particularly important problem is to ensure the quality of service of large volumes of heterogeneous network traffic in conditions of flow and network environment uncertain dynamic. Therefore, among existing means of QoS basic architecture the special role is played by the mechanisms of network resources management and data transmission processes [1, 2].

Such network share as the router buffer (queue) size is common for any packets class. The overloading of buffer and resulting increase in packet loss occurring in the router lead to deterioration of QoS main parameters which is unacceptable for some classes of traffic. Therefore, the actual problem is the development packet loss reduction methods to improve the router buffer management. From a practical point of view such QoS methods are interesting because they do not require additional expenses or any modernization of the existing network infrastructure.

Given the above, it was decided to explore opportunities to improve QoS parameters through the operating system MikroTik RouterOS, which is used in the routers of the producer. The TCP/IP was chosen as network architecture because the protocols of this family are the main protocols, which provide control and routing messages over a data network.

The work is devoted on making simple, cheap and effective method of QoS improvement. To do this, we need to analyze the advantages and disadvantages of router with RouterOS operating system standard operating algorithm, identify the parameters of the algorithm that can be affected in order to improve its efficiency, modify the standard algorithm and perform an experimental test of its efficiency.

2. Aim

The aim of this work is to develop methods to reduce packet loss regardless of the traffic type (file, media, web-traffic, etc.) and the general improvement of the quality of the network service through the use of standard tools of OS RouterOS. To achieve this goal the following tasks are identified:

- analysis of existing solutions and identification of gaps;
- development of algorithms to eliminate deficiencies in the selected method;
- software installation and configuration (establishing a separate virtual machine MikroTik RouterOS; establishing two virtual machines at Windows XP; setting communication between two Windows XP via MikroTik RouterOS);
- writing a sequence of commands that implement the algorithm and add it to MikroTik RouterOS;
- conducting the algorithm tests;
- analyzing results.

3. Main results

VMware Workstation 10 software was chosen to simulate the network. Three insulated from each other virtual machines were created. Two of them contained Microsoft Windows XP and at the third one - RouterOS. Both Windows XP are the network resources users, their task was to upload in parallel way an arbitrary information from the Internet to use the available bandwidth [3]. The RouterOS is a specialized mini-operating system that uses PC resources to turn it into a multifunctional router which was used for bandwidth regulation.

A special algorithm was developed to solve the problem of bandwidth regulation, the main features of which are:

- perform all the standard functions (providing guaranteed speed for clients in all conditions and the ability to increase the speed to specified limit);
- multilevel analysis of traffic (for groups of users, types of traffic by priority)
- implementation of double QoS approach [4, 5] (previous labeling of all incoming traffic as a low priority and then re-marking a traffic that should have a higher priority after analysis);
- Assignment of medium priority to packets that are not under any rule in the list.

A block diagram of the algorithm is presented at Fig. 1. The basic configuration of the algorithm was performed by using console commands of OS RouterOS. To implement the algorithm into a MikroTik router the following procedure must be performed: users groups were created; records in the IP - Firewall (tab Layer7 Protocols) were added; an improved system of rules in the IP - Firewall (tab Mangle) was established; more templates for queues in the Queue (tab Queue Types) were created; the queues tree in the Queue (tab Queue Tree) was created using the Mangle rules and the Queue Types templates.

The algorithm involves five groups: A, B, C, D and E. Each of them contains four classes (A, B, C, D) in which traffic priorities are gradually declining (from 1 to 8). For example, the Group A includes traffic priorities from 1 to 4, Group B - from 2 to 5, etc. The groups are created by commands shown below:

```
/ip firewall address-list
add address=192.168.0.1 disabled=no list=GROUP-A
add address=192.168.0.5 disabled=no list=GROUP-E
```

The Layer7 Protocols is a method of determining the connection parameters type of ICMP/TCP/UDP flows. Properly compiled templates (called the regular expressions) for these protocols are able to manage very flexibly by all traffic passing through the router. Regular expression is a clear set of characters that uniquely define the connection type (client application, system service protocols that use TCP-connection, HTTP, etc.). In fact, regular expressions are patterns (masks) by which is conducted the search of bytes sequence in first ten packets or 2 KB each established connection. If there is no match, then further search is terminated and the connection is considered unknown. If a match is found, then on tabs Filter Rules, NAT, or Mangle it is possible to create the rules chains with necessary actions to be executed with this type of traffic.

There are two ways to define the mask properly:

- using the regular expressions lists from the Internet;
- getting first 2 KB of connection, find the necessary sequence, using, for example Wireshark, and transforming this sequence into template at the specialized site.

There are special rules to standardize the regular expressions syntax. Here's how to add the regular expressions to identify connection using commands of operating system RouterOS:

```
/ip firewall layer7-protocol
add name=Skype regexp="^..\\x02............."
add name=GIF_FILE regexp=gif
```

Marking connections and packets are being implemented in IP - Firewall (tab Mangle). Here the main work on multi-traffic analysis and the double QoS implementation is executed.

The very first step is pre-marking of all incoming packet stream (first level QoS). The lowest priority is assigned to it. This is made in order to make possible briefly delaying the incoming traffic for its analysis and subsequent separation of traffic types with a higher priority.

Providing the lowest priority to input traffic in the router console looks like this:

```
/ip firewall mangle
add action=mark-connection chain=forward comment=ALLTRAFFIC disabled=no new-connection-mark=CLASS-D passthrough=yes
add action=mark-packet chain=forward comment=CLASS-D-GROUP-E-DL connection-mark=CLASS-D disabled=no dst-address-list=GROUP-E new-packet-mark=CLASS-D-GROUP-E-DL passthrough=yes
```

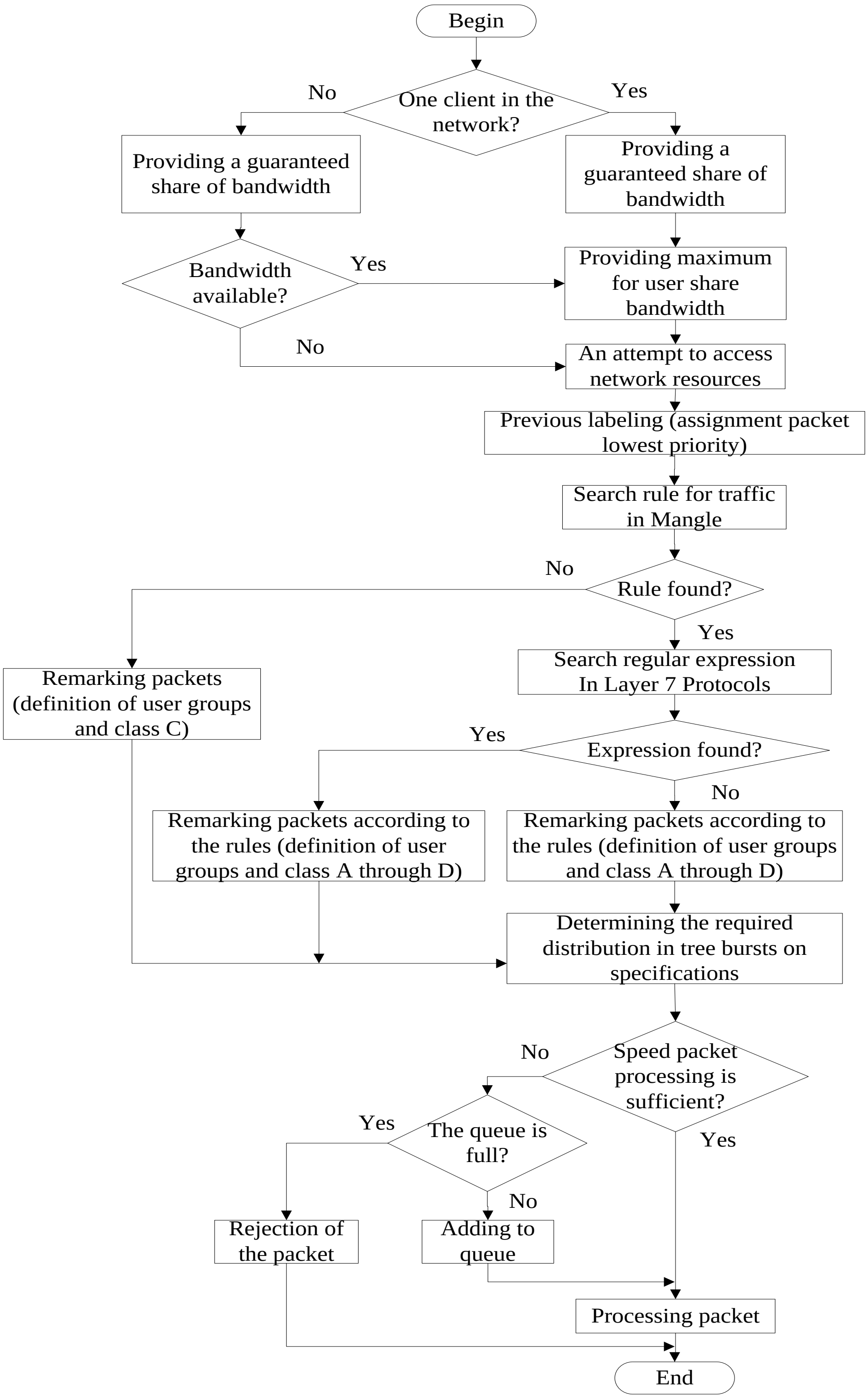

Begin
No
One client in the network?
Yes
Providing a guaranteed share of bandwidth
Providing a guaranteed share of bandwidth
Bandwidth available?
Yes
Providing maximum for user share bandwidth
No
An attempt to access network resources
Previous labeling (assignment packet lowest priority)
Search rule for traffic in Mangle
No
Rule found?
Yes
Remarking packets (definition of user groups and class C)
Search regular expression In Layer 7 Protocols
Yes
Expression found?
No
Remarking packets according to the rules (definition of user groups and class A through D)
Remarking packets according to the rules (definition of user groups and class A through D)
Determining the required distribution in tree bursts on specifications
No
Speed packet processing is sufficient?
Yes
The queue is full?
Yes
No
Rejection of the packet
Adding to queue
Processing packet
End

Figure.1 Block diagram of the modified traffic management algorithm

Received: March 17, 2017

Further search of rules in the module Mangle rule list is conducted according to the parameters of connection (IP-source address, IP-destination address, source port, destination port). If there is no special rule, the group to which it belongs is determined by the user's IP-address, the class C traffic and the corresponding priority (second level QoS) are assigned to the packets and the flow is forwarded to the global queue. If a rule in Mangle is found, an additional search in list Layer7 Protocols for more accurate determination of connection parameters and type of traffic to be transmitted is conducted. If a regular record is found in the list, the router also re-marks all packets, but assigns classes listed in the rule (the second level QoS) from A (highest) to D (lowest). The user group priorities are specified and the flow is redirected to the global queue. The rule specifying the type of connection described in Layer7 Protocols follows below:

```
/ip firewall mangle
add action=mark-connection chain=forward comment=Skype disabled=no layer7-protocol=Skype new-connection-mark=CLASS-B passthrough=yes
```

If there are no matches in the list of templates, the re-marking is performed as it is in the previous case. Using the IP-address to determine the user groups, the classes from A to D and corresponding priorities are assigned to traffic packets and the flow is forwarded to the global queue. The rule without precise instructions about connection type is as follows:

```
/ip firewall mangle
add action=mark-connection chain=forward comment=FTP disabled=no dst-port=20,21 new-connection-mark=CLASS-C passthrough=yes protocol=tcp
add action=mark-connection chain=forward comment=ICQ disabled=no dst-port=5190 new-connection-mark=CLASS-B passthrough=yes protocol=tcp
```

The global queue does not store the packets and forwards them according to marking to sub queues (if any). Also global queue establishes the general maximum speed limits for all sub queues.

At the second level of the hierarchy tree the clients are divided into groups. Depending on the marking the packet can get into different groups. Since this is not the last level of the hierarchy, it also does not store the packets but sorts them according to the markings. At this level the limitation of the guaranteed and maximum speeds for all sub queues is set as well.

At the last level definition of class priority for the group is made and packets are stored while transferred to the output interface. The creation of such queues in the module Queue - Queue tree is given below:

```
/queue tree
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-at=0 max-limit=0 name=CLASS-DE-DL packet-mark=CLASS-D-GROUP-E-DL priority=8 parent=GROUP-E-DL queue=GROUP-E-DL
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-at=0 max-limit=0 name=CLASS-CB-DL packet-mark=CLASS-C-GROUP-B-DL priority=4 parent=GROUP-B-DL queue=GROUP-B-DL
```

The established rules and the queue tree with specified guaranteed and maximum speeds are shown in Fig. 2.

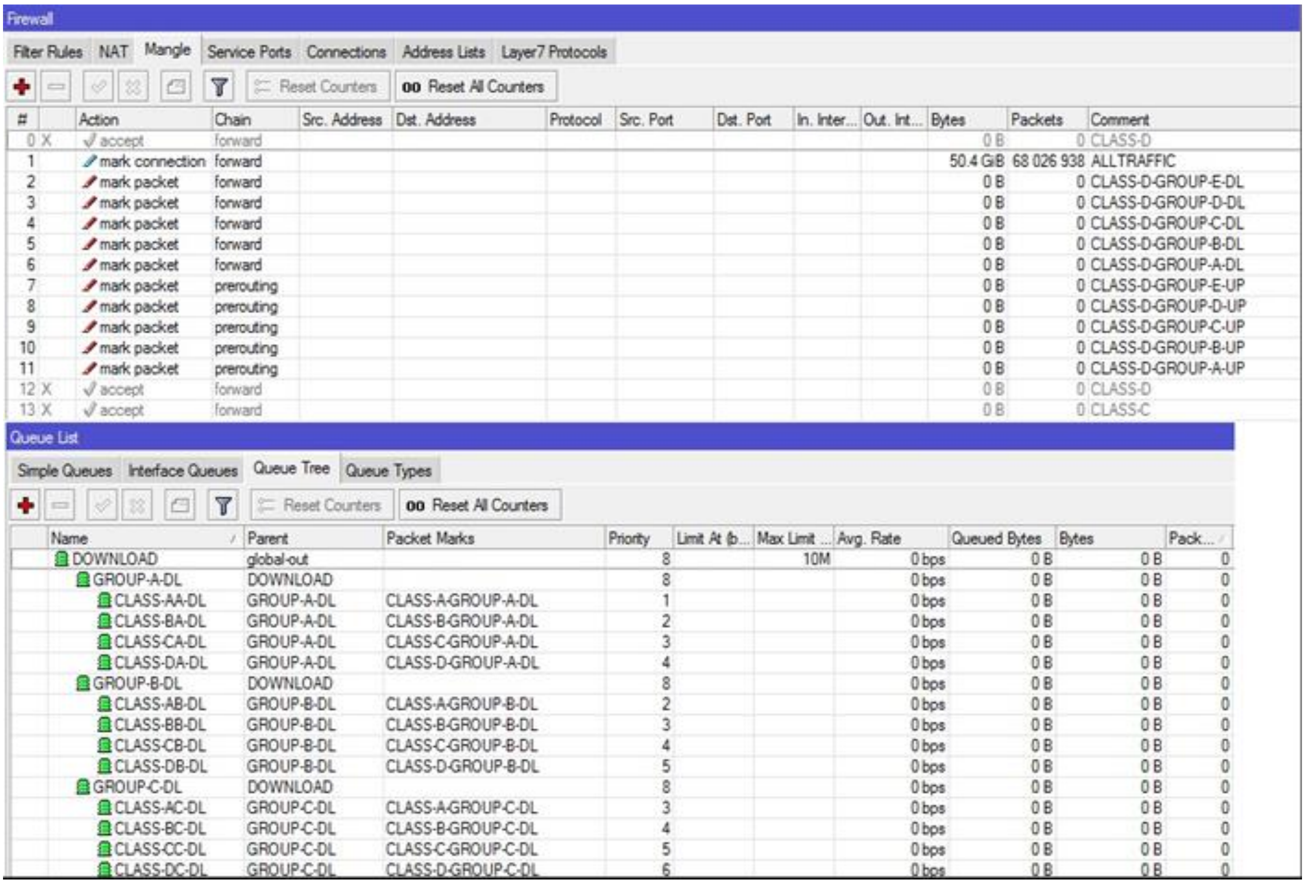
The algorithm test was performed in three phases:

- creating conditions for overloading of network, controlled by a router with a standard algorithm, counting the number of sent and lost packets and building graphs;
- creating similar conditions in the network controlled by a router with the developed algorithm, counting the number of sent and lost packets and building graphs;
- comparing the two algorithms results.

The results of the first phase of network test are presented in Table 1. Fig. 3 shows the basic parameters of the first client test and Fig. 4 shows those for the second one.

The results of the second phase of test are given in Table 2. Fig. 5 shows the diagrams of the basic parameters of the test conducted for the first client, and Fig. 6 shows those for the second one.

The D-ITG (Distributed Internet Traffic Generator) packet was used to generate traffic with a given distribution of the two parameters: Packet Size (PS) and Internet Departure Time (IDT) [6]. In this study we used the uniform distribution law and the following ranges of parameters: PS = 512…1024 bytes, IDT = 20…100 packets/sec in multi-traffic generation mode (Custom, Voice, DNS, Telnet, Gaming).



#		Action	Chain	Src. Address	Dst. Address	Protocol	Src. Port	Dst. Port	In. Inter...	Out. Int...	Bytes	Packets	Comment
0	X	accept	forward								0 B	0	CLASS-D
1		mark connection	forward								50.4 GiB	68 026 938	ALLTRAFFIC
2		mark packet	forward								0 B	0	CLASS-D-GROUP-E-DL
3		mark packet	forward								0 B	0	CLASS-D-GROUP-D-DL
4		mark packet	forward								0 B	0	CLASS-D-GROUP-C-DL
5		mark packet	forward								0 B	0	CLASS-D-GROUP-B-DL
6		mark packet	forward								0 B	0	CLASS-D-GROUP-A-DL
7		mark packet	prerouting								0 B	0	CLASS-D-GROUP-E-UP
8		mark packet	prerouting								0 B	0	CLASS-D-GROUP-D-UP
9		mark packet	prerouting								0 B	0	CLASS-D-GROUP-C-UP
10		mark packet	prerouting								0 B	0	CLASS-D-GROUP-B-UP
11		mark packet	prerouting								0 B	0	CLASS-D-GROUP-A-UP
12	X	accept	forward								0 B	0	CLASS-D
13	X	accept	forward								0 B	0	CLASS-C

Name	Parent	Packet Marks	Priority	Limit At (b...	Max Limit ...	Avg. Rate	Queued Bytes	Bytes	Pack...
DOWNLOAD	global-out		8		10M	0 bps	0 B	0 B	0
GROUP-A-DL	DOWNLOAD		8			0 bps	0 B	0 B	0
CLASS-AA-DL	GROUP-A-DL	CLASS-A-GROUP-A-DL	1			0 bps	0 B	0 B	0
CLASS-BA-DL	GROUP-A-DL	CLASS-B-GROUP-A-DL	2			0 bps	0 B	0 B	0
CLASS-CA-DL	GROUP-A-DL	CLASS-C-GROUP-A-DL	3			0 bps	0 B	0 B	0
CLASS-DA-DL	GROUP-A-DL	CLASS-D-GROUP-A-DL	4			0 bps	0 B	0 B	0
GROUP-B-DL	DOWNLOAD		8			0 bps	0 B	0 B	0
CLASS-AB-DL	GROUP-B-DL	CLASS-A-GROUP-B-DL	2			0 bps	0 B	0 B	0
CLASS-BB-DL	GROUP-B-DL	CLASS-B-GROUP-B-DL	3			0 bps	0 B	0 B	0
CLASS-CB-DL	GROUP-B-DL	CLASS-C-GROUP-B-DL	4			0 bps	0 B	0 B	0
CLASS-DB-DL	GROUP-B-DL	CLASS-D-GROUP-B-DL	5			0 bps	0 B	0 B	0
GROUP-C-DL	DOWNLOAD		8			0 bps	0 B	0 B	0
CLASS-AC-DL	GROUP-C-DL	CLASS-A-GROUP-C-DL	3			0 bps	0 B	0 B	0
CLASS-BC-DL	GROUP-C-DL	CLASS-B-GROUP-C-DL	4			0 bps	0 B	0 B	0
CLASS-CC-DL	GROUP-C-DL	CLASS-C-GROUP-C-DL	5			0 bps	0 B	0 B	0
CLASS-DC-DL	GROUP-C-DL	CLASS-D-GROUP-C-DL	6			0 bps	0 B	0 B	0

Figure.2 Queues hierarchy of the created algorithm

Table 1. Results of testing network, controlled by router with a standard algorithm

Test number	Standard algorithm for Win XP 1		Standard algorithm for Win XP 2	
	Total number of packets	Packets lost	Total number of packets	Packets lost
1	323085	151261	67853	22664
2	82075	29819	63474	20831
3	202580	70540	65663	30747
4	217275	95532	209227	95038
5	29762	3112	21247	10410
6	123518	26772	127190	46882
7	144413	20460	146155	63133
8	284524	98686	52971	22637
9	214468	50073	166316	70306
10	15101	2703	27145	6850
11	318732	127910	54344	19125
12	166316	16306	123518	47937
13	203152	75155	126888	52713
14	51229	12609	14679	2228
15	127190	26882	70783	27470
Average value	166894	53855	89163	35931

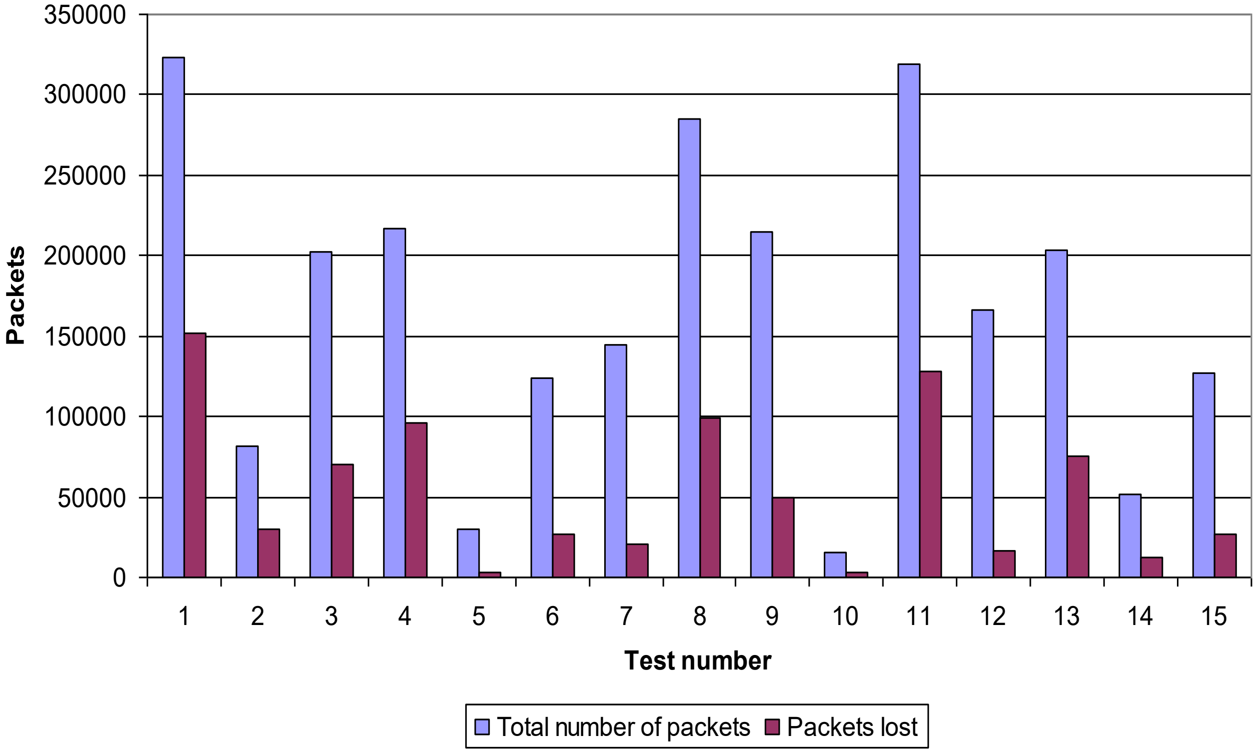


Figure.3 Diagram of the basic parameters of the first phase of test for the first client (Win XP 1)

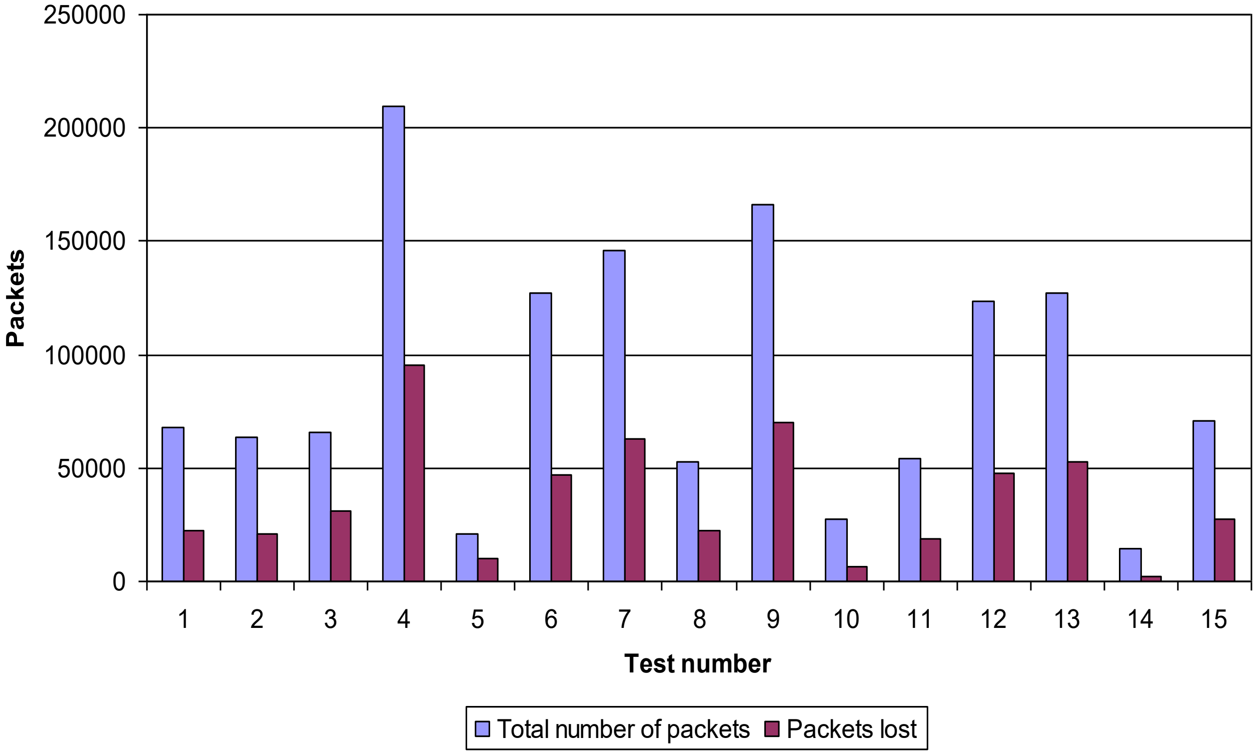


Figure.4 Diagram of the basic parameters of the first phase of test for the second client (Win XP 2)

Table 2. Network testing results under control of the developed algorithm

Test number and traffic class		Standard algorithm for Win XP 1				Standard algorithm for Win XP 2			
		Total number of packets		Packets lost		Total number of packets		Packets lost	
1	A	145542	145597	29962	29967	95480	95744	5166	5183
	B	55		5		165		17	
	C	0		0		99		0	
2	A	235689	235721	71866	71866	242944	243127	94509	94509
	B	32		0		154		0	
	C	0		0		29		0	
3	A	28209	29367	1768	1939	332419	332422	154377	154377
	B	190		0		3		0	
	C	968		171		0		0	
4	A	140516	140534	41019	41019	174212	174346	48437	48437
	B	18		0		110		0	
	C	0		0		24		0	
5	A	102074	102104	31134	31134	242767	242767	76776	76776
	B	30		0		0		0	
	C	0		0		0		0	
6	A	291307	292157	127407	127531	205115	205840	39429	39472
	B	602		0		191		0	
	C	248		124		534		43	
7	A	82753	83031	19337	19337	212131	212131	46862	46862
	B	106		0		0		0	
	C	172		0		0		0	
8	A	21554	21938	2194	2213	230178	230251	54789	54789
	B	188		0		73		0	
	C	196		19		0		0	
9	A	70689	71146	22863	22946	230975	231004	73510	73510
	B	37		0		29		0	
	C	420		83		0		0	
10	A	138311	138533	27245	27353	226020	226286	22998	23012
	B	107		0		88		0	
	C	115		108		178		14	
11	A	16190	24864	981	1509	170686	170782	65533	65533
	B	286		0		96		0	
	C	8388		528		0		0	
12	A	82197	82197	17022	17022	70928	70978	4982	4982
	B	0		0		50		0	
	C	0		0		0		0	
13	A	89603	89703	15071	15071	123941	124013	14821	14821
	B	43		0		58		0	
	C	57		0		14		0	
14	A	119772	119920	34731	34737	238629	238715	54358	54358
	B	83		0		76		0	
	C	65		6		10		0	
15	A	170068	170689	55946	56076	256169	256421	105772	105786
	B	109		0		74		0	
	C	512		130		178		14	
Average value	A	115632	116500	33236	33315	203506	203655	57488	57494
	B	126		0		78		1	
	C	743		78		71		5	

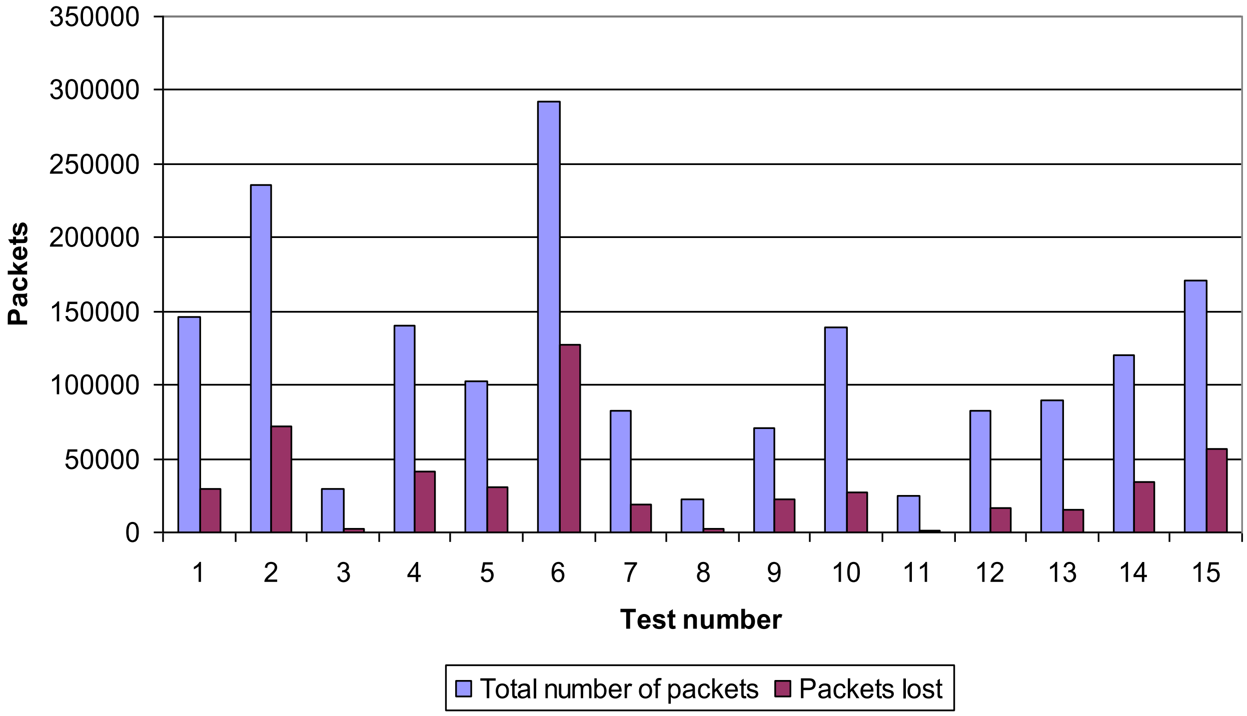


Figure.5 Diagram of the basic parameters of the second phase of test for the first client (Win XP 1)

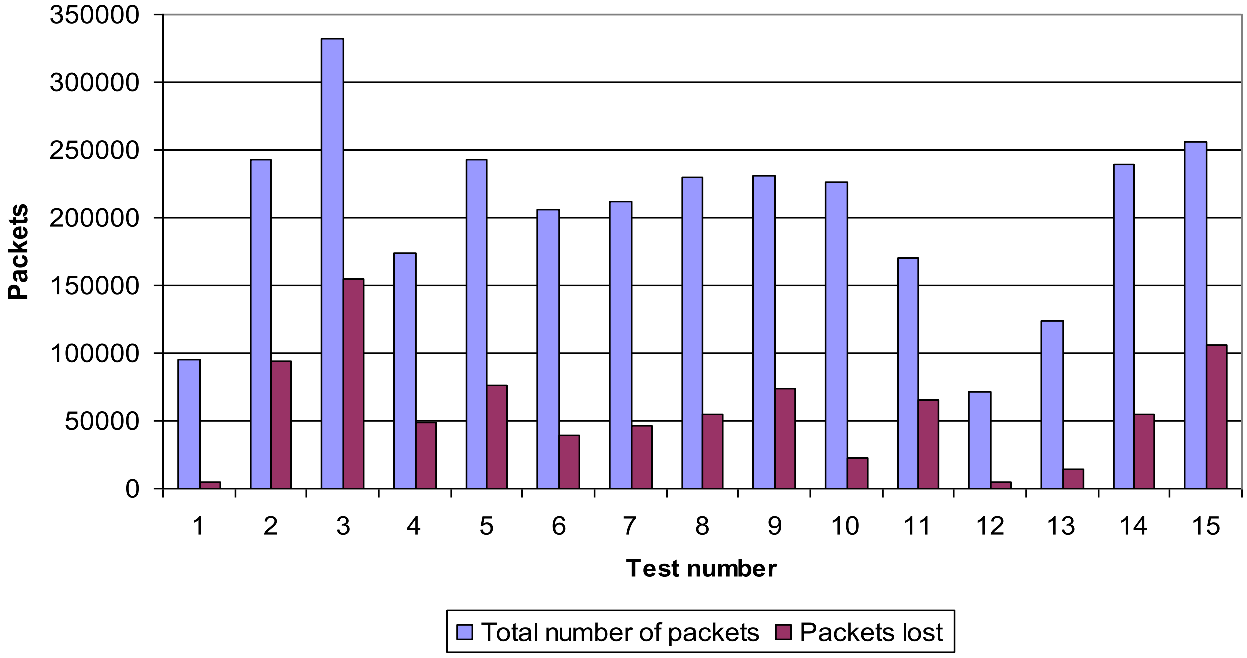


Figure.6 Diagram of the basic parameters of the second phase of test for the second client (Win XP 2)

For the last step it is necessary to calculate the average number of packets that have passed through the queue and discarded packets across the network for standard and developed control router algorithms and to build a summary chart of these parameters (Fig. 7).

For a standard algorithm:

(166 894 + 89 163) / 2 = 128 029 (packets passed through the queue);

(53 855 + 35 931) / 2 = 44 893 (dropped packets).

For the developed algorithm:

(116 500 + 203 655) / 2 = 160 078 (packets passed through the queue)

(33 315 + 57 494) / 2 = 45 405 (dropped packets)

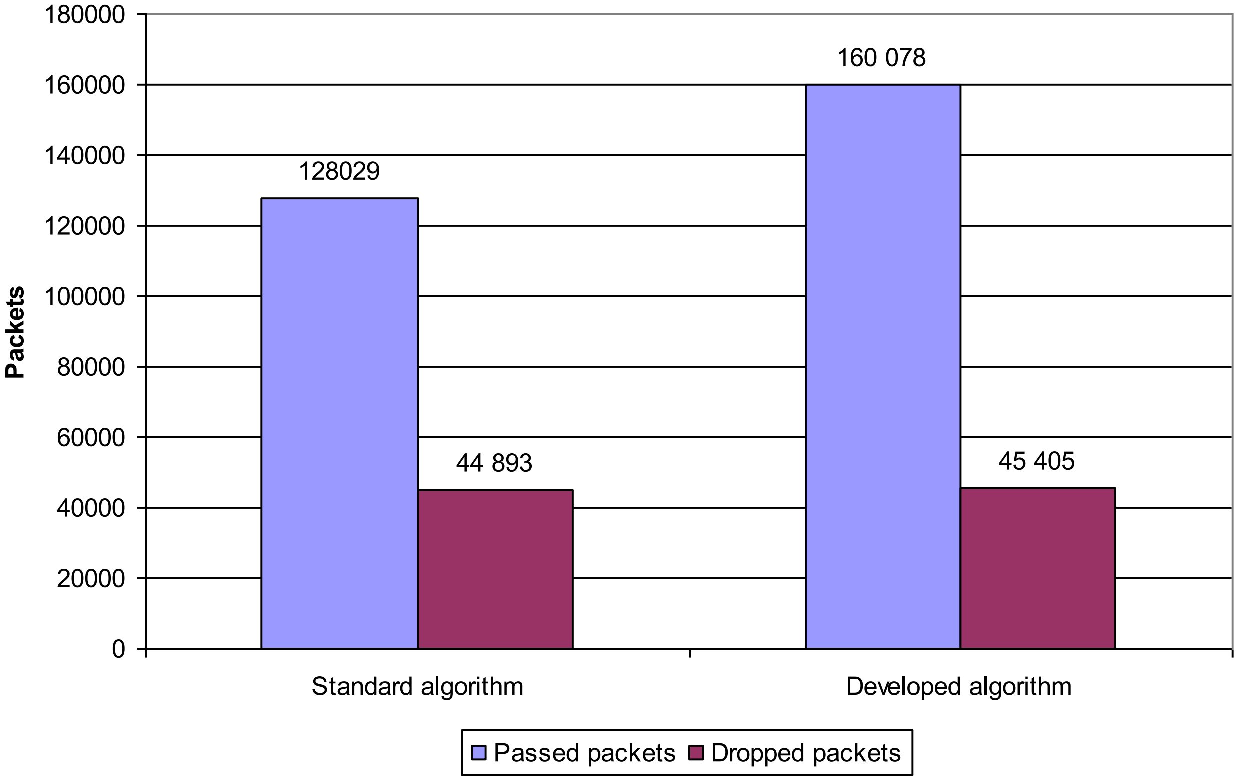


Figure.7 Comparison chart of mean parameters values of standard and developed algorithms

To determine the developed algorithm effectiveness, the percentage of discarded packets compared with the total number of packets can be calculated:

44 893 · 100% / 128 029 = 35.06%;
45 405 · 100% / 160 078 = 28.36%.

We can see that implementation of the standard algorithm resulted in loss of 35.06% of all packets, while implementation of the developed algorithm resulted in loss of 28.36% packets. Thus the developed algorithm is 6.7% more effective than the standard one in adopted circumstances.

4. Conclusions

The paper suggests an improved double QoS method for Mikrotik routers which differs from standard by application of multilevel traffic analysis of users groups, traffic types and priorities. In experiments conducted for evenly distributed traffic of 512...1024 byte packet length and intensity of 20…100 packets/sec compared with a standard algorithm the number of lost packets has been reduced by 6.7%.

Thus the experiment proved that the application of the proposed method reduces the number of discarded packets. This method can be proposed to improve the efficiency of existing network equipment.

In future studies it is planned to implement the proposed method using the local model of the controlled process conception [7], which will solve the problem of improving the quality of service in conditions of structural and parametric uncertainty of network and traffic.

References

[1] O.G. Slavko, "Skhema adaptyvnogho zabezpechennja QoS iz vykorystannjam modyfikovanogho metodu ARED" [Scheme of QoS adaptive providing using the modified ARED method], *Journal of Kremenchuk Mykhailo Ostrogradskyi National University*, Vol. 3, No. 80, pp. 107-112, 2013.

[2] O.G. Slavko "Adaptive Control Method of Network Data Flow for QoS", In: *Proc. of the 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Berlin, Germany, Vol. 1, pp. 521-524, 2013.

[3] M. Guchenko, N. Sokhin, and O. Lugovyj "Local computer networks simulation at the base of virtual machines technology in QoS tasks", In: *Proc. of Scientific and Student's Works in the Field of Industrial Electrical*

Engineering, Košice, Slovakia, pp. 148-150, 2016.

[4] A. Giordano, “QoS in RouterOS *v6.x*”, *MikroTik User Meeting*, 2014. [Online]. Available: http://mum.mikrotik.com/presentations/IT14/giordano.pdf. [Accessed: 07-Nov.-2016].

[5] “Double QoS for v6 is possible”, 2012. [Online]. Available: http://forum.mikrotik.com/viewtopic.php?t=67037. [Accessed: 05-Nov.-2016].

[6] A. Botta, W. de Donato, A. Dainotti, S. Avallone, and A. Pescape, “*D-ITG 2.8.1 Manual, 2013*”, *COMICS*, 2013. [Online]. Available : http://www.grid.unina.it/software/ITG/manual/D-ITG-2.8.1-manual.pdf. [Accessed: 07-Nov.-2016].

[7] M. Guchenko, O. Slavko, J. Olkhova, P. Kostenko, and M. Ivanova “Simulation of a Control System Based on a Local Model of a Controlled Process”, In: *Proc. of the 1st Slovak-Ukrainian conference of young scientists*, Banska Bystrica, Slovakia, pp. 51-56, 2010.